Indiana Department of Education

Academic Standards Course Framework

COMPUTER PROGRAMMING II

Computer Programming II explores and builds skills in C++ and Java. The study of C++ provides a basic understanding of the fundamentals of procedural program development using structured, modular concepts. Emphasizes logical program design involving user-defined functions and standard structure elements. Discussions will include the role of data types, variables, structures, addressable memory locations, arrays and pointers. Data file access methods are also presented. The development of Java programming skills will provide a basic understanding of the fundamental concepts with an emphasis on logical program design using a modular approach which involves task oriented program functions. Java allows the design of an Internet user interface. The application is built by selecting forms and controls, assigning properties and writing code.

- DOE Code: 5236
- Recommended Grade Level: Grade 11-12
- Recommended Prerequisites: Computer Programming I
- Credits: 1-3 credits per semester, maximum of 6 credits
- Counts as a Directed Elective or Elective for the General, Core 40, Core 40 with Academic Honors and Core 40 with Technical Honors diplomas
- This course is aligned with postsecondary courses for Dual Credit:
  - o Ivy Tech
    - CINS 121 C++ Programming
    - CINS 136 Java Programming
  - o Vincennes University
    - COMP 203 C++ Programming
    - COMP 252 Java Programming

Dual Credit
This course provides the opportunity for dual credit for students who meet postsecondary requirements for earning dual credit and successfully complete the dual credit requirements of this course.

Application of Content and Multiple Hour Offerings
Intensive laboratory applications are a component of this course and may be either school based or work based or a combination of the two. Work-based learning experiences should be in a closely related industry setting. Instructors shall have a standards-based training plan for students participating in work-based learning experiences. When a course is offered for multiple hours per semester, the amount of laboratory application or work-based learning needs to be increased proportionally.

Career and Technical Student Organizations (CTSOs)
Career and Technical Student Organizations are considered a powerful instructional tool when integrated into Career and Technical Education programs. They enhance the knowledge and skills students learn in a course by allowing a student to participate in a unique program of career and leadership development. Students should be encouraged to participate in Business Professional of America, DECA, or Future Business Leaders of America, the CTSOs for this area.

Content Standards

Domain – Task Analysis

Core Standard 1 Students evaluate the tasks a computer program is to perform.

    Standards

    CPII-1.1   Differentiate between the different tasks a computer program should perform

    CPII-1.2   Formulate solutions to the different tasks a computer program should perform

    CPII-1.3   Interpret prior solutions. In case prior code and design could be reused

Domain – Problem Analysis

Core Standard 2 Students analyze a problem and develop an advanced solution by creating a computer program.

    Standards

    CPII-2.1   Recognize and explain how to use a computer program to solve a problem

    CPII-2.2   Construct interactive computer programs that accept various forms of input and produce various forms of output, as a solution to an advanced computer programming problem

    CPII-2.3   Use print charts, file layouts, program narratives, hierarchy charts, and system flowcharts, which accurately depict the problem assigned and describe the solution

    CPII-2.4   Justify what programming methodology to use—object oriented or procedural

    CPII-2.5   Appraise the program schematics and usage; document the program and describe its use

    CPII-2.6   Recognize and explain the standard program flowchart symbols and use them correctly within the context of the basic control structures of sequence, selection and looping

Domain – Software Tools

Core Standard 3 Students apply and adapt software tools to develop an advanced computer program.

    Standards

    CPII-3.1   Construct an advanced program that processes information

    CPII-3.2   Identify programming languages as procedural and object oriented

    CPII-3.3   Recognize and explain class in object oriented programming

    CPII-3.4   Recognize and explain object in object oriented programming

    CPII-3.5   Recognize and explain method in object oriented programming

    CPII-3.6   Recognize and explain instance variable in object oriented programming

    CPII-3.7   Recognize and explain polymorphism in object oriented programming

    CPII-3.8   Recognize and explain inheritance in object oriented programming

    CPII-3.9   Recognize and explain overwriting methods in OOP

    CPII-3.10 Recognize and explain encapsulation in computer programming

    CPII-3.11 Apply and adapt at an advanced level fundamental programming concepts, including data types, control structures, methods, and arrays

    CPII-3.12 Develop advanced programs using reusable modules (modularization)

    CPII-3.13 Use advanced debugging techniques to correct and validate the computer program

    CPII-3.14 Construct the program in a high-level programming language based on a created design

    CPII-3.15 Determine how to integrate a computer program with a web browser

    CPII-3.16 Determine how to use a common code/ GUI library

CPII-3.17 Identify controls (push buttons, entry fields, etc.), their properties, methods, and when to use each control

Domain – Algorithms

Core Standard 4 Students design a solution to the problem using algorithms.

Standards

CPII-4.1   Develop advanced algorithms to solve a computer programming problem(s)

CPII-4.2   Apply and adapt math operators in a computer program

CPII-4.3   Prescribe the use of algorithms to provide a solution to a programming problem

CPII-4.4   Use pseudo code to describe a solution to an advanced programming problem

CPII-4.5   Create a program flowchart and ANSI standard flowcharting symbols to define a solution to an advanced programming problem

CPII-4.6   Explain how the algorithm can be used to solve a problem

Domain – Program Development

Core Standard 5 Students create an advanced functional computer program.

Standards

CPII-5.1   Define the process of programming. For example STAIR, Statement, Tools, Algorithm, Implement, and Refine

CPII-5.2   Create an advanced computer program that corresponds to an algorithm or proposed solution

CPII-5.3   Demonstrate programming structures

CPII-5.4   Appraise the use of data variables and constants

CPII-5.5   Appraise the use of local scope and global scope

CPII-5.6   Appraise the use of conditionals (IF statements)

CPII-5.7   Appraise the use of loops (while statements, for statements)

CPII-5.8   Use single and multidimensional Arrays

CPII-5.9   Create programmer defined functions and methods to break down the program logic and support reuse

CPII-5.10 Define the graphical user interface

CPII-5.11 Identify the parts of the programming platform

CPII-5.12 Identify different types of errors and handle them programmatically

CPII-5.13 Use the order of operations when using calculations

CPII-5.14 Construct an advanced computer program using proper condition and loop techniques

CPII-5.15 Use correct naming conventions in variable declarations, function declarations, class declarations, and other. Camel casing

Domain – Program Verification and Debugging

Core Standard 6 Students prove that an advanced computer program solution works by using verification and debugging techniques.

Standards

CPII-6.1   Predict and explain output

CPII-6.2   Identify cause/effect for input/output

CPII-6.3   Perform input validation

CPII-6.4   Scrutinize peers code for errors

CPII-6.5   Show the use of proper internal documentation and coding comments

Domain – Documentation

Core Standard 7 Students connect the associated task with the code by providing documentation.

Standards

CPII-7.1   Describe the function of an advanced computer program

CPII-7.2   Identify the purposes of an advanced computer program

CPII-7.3   Explain concepts related to an advanced computer program

CPII-7.4   Evaluate how to use an advanced computer program

CPII-7.5   Identify cause/effect by explaining input and output related to an advanced computer program

CPII-7.6   Interpret input/output of an advanced computer program

Career and Technical Student Organizations

Career and Technical Student Organizations are considered a powerful instructional tool when integrated into Career and Technical Education programs. They enhance the knowledge and skills students learn in a course by allowing a student to participate in a unique program of career and leadership development. Students should be encouraged to participate in a Career and Technical Student Organization, such as Business Professional of America, DECA, or Future Business Leaders of America.